User interface design

Week10: Prototyping and construction: prototyping, conceptual design, Concrete design, generating prototypes, construction

Nagulama Moses

Lecturer

Department Of Information Technology

Kumi University

Email: mnagulama@gmail.com

Outline

- Intended learning outcomes
- Prototyping
- Conceptual design
- Concrete design
- Generating prototypes
- Construction

Intended learning outcomes

- Able to identify the different types of prototypes, including physical, digital, and paper prototypes.
- Able to describe prototyping and the different types of prototyping activities.
- Able to enable one to produce prototypes from the models developed during the requirements activity.
- Enable one to produce a conceptual model for a product and justify your choices.

Introduction

- Design, prototyping, and construction fall within the Develop phase of the double diamond of design in which solutions or concepts are created, prototyped, tested, and iterated.
- The final product emerges iteratively through repeated design-evaluation-redesign cycles involving a range of stakeholders, and prototypes facilitate this process.
- There are two aspects to design: a conceptual part, which focuses on ideas for a product, and the concrete part, which focuses on the details of the design.

- The conceptual part involves developing a conceptual model that captures what the product will do and how it will behave, while the concrete part is concerned with the details of the design, such as menu types, haptic feedback, physical widgets, and graphics.
- The two are intertwined, as concrete design issues will require some consideration in order to prototype ideas, and prototyping ideas will lead to an evolution of the concept.

- Designers prototype their design ideas so that people can evaluate them effectively.
- In the early stages of development, these prototypes may be made of paper and cardboard or be ready-made components pulled together to allow evaluation, while as the design progresses, they become more polished, tailored, and robust so that they resemble the final product.

Prototyping

- Prototyping provides a concrete manifestation of an idea whether it is a new product or a modification of an existing one which allows designers to communicate their ideas and for others to try them out.

What Is a Prototype?

- A prototype is one manifestation of a design that allows stakeholders to interact with it and to explore its suitability.
- It is limited in that a prototype will usually emphasize one set of product characteristics and de-emphasize others.
- Prototypes take many forms, for example, a scale model of a building or a bridge, or a piece of software with limited capabilities.

- A prototype can also be a paper-based outline of a display, a collection of wires and ready-made components, a digital picture, a video simulation, a complex piece of software and hardware, or a three-dimensional mockup of a workstation.
- In fact, a prototype can be anything from a paper-based storyboard to a complex piece of software and from a cardboard mockup to a molded or pressed piece of metal.
- For example, when the idea for the PalmPilot (a predecessor to mobile and smartphones) was being developed, Jeff Hawkins (founder of the company) carved up a piece of wood about the size and shape of the device he had imagined.

Figure 1: The PalmPilot wooden prototype (Yvonne Rogers, 2023, P.444)

- Jeff Hawkins used to carry this piece of wood around with him and pretend to enter information into it, just to see what it would be like to own such a device.
- This is an example of a simple prototype, but it served its purpose of simulating scenarios of use.
- Advances in 3D printer technologies, coupled with reduced prices, have increased their use in design.
- It is now common practice to take a 3D model from a software package and print a prototype, or indeed a final product.

Why Prototype?

- Prototypes are useful when discussing or evaluating ideas with stakeholders; they are a communication device among team members and an effective way for designers to explore design ideas.
- The activity of building prototypes encourages reflection in design, and it is recognized by designers from many disciplines as an important aspect of design.
- Prototypes answer questions and support designers in choosing between alternatives.

- Prototypes test the technical feasibility of an idea, to clarify some vague requirements, to do some evaluation, or to check that a certain design direction is compatible with the rest of product development.
- Prototypes may also be deployed as probes, and can be the focus for a wider exploration of future technologies.
- It encourages the product team to view the idea differently by transforming it into a more realized form than just a concept.

Example of prototypes

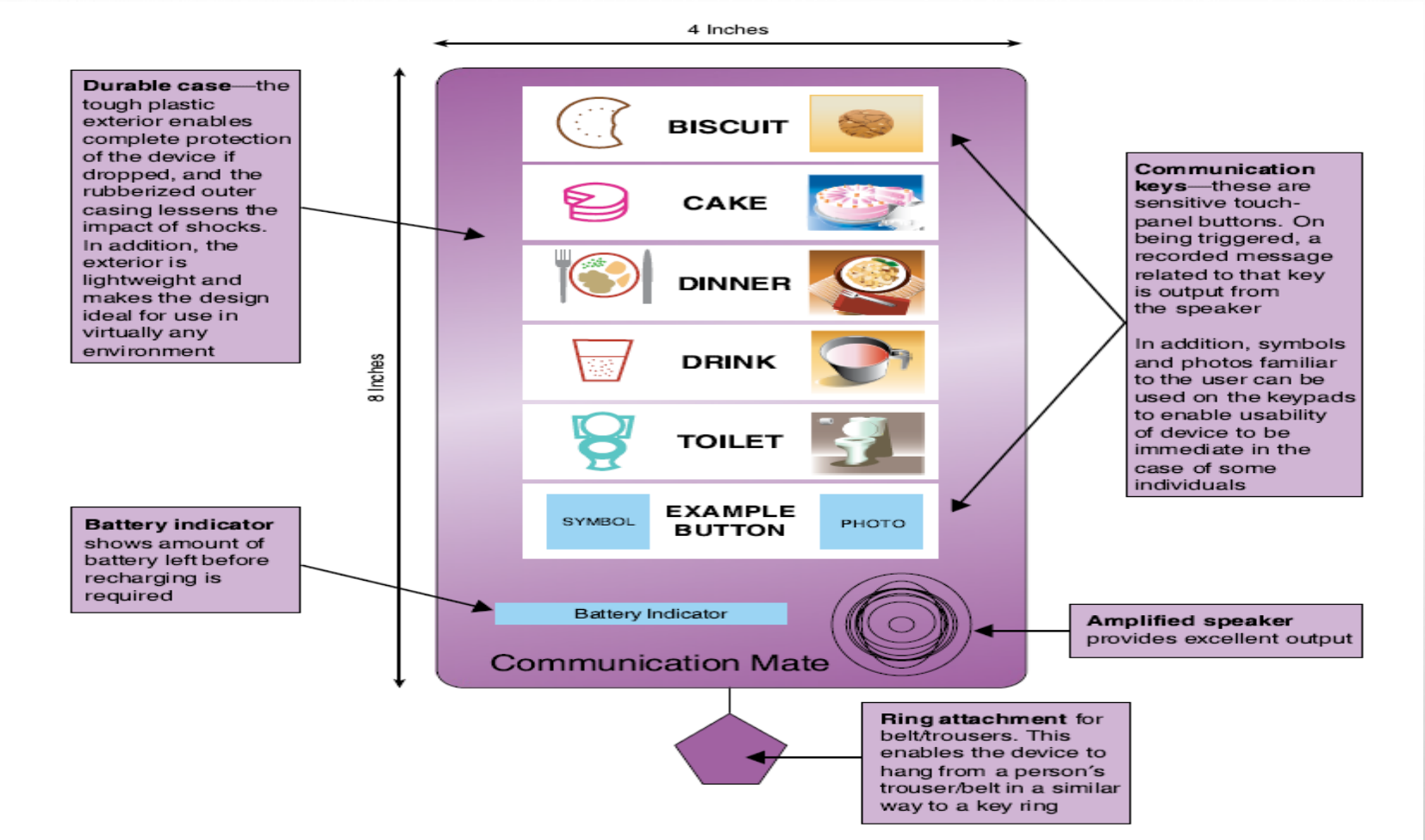

Figure 2: A paper-based prototype of a handheld device to support an autistic child (Yvonne Rogers, 2023, P.446)

- This prototype shows the intended functions and buttons, their positioning and labeling, and the overall shape of the device, but none of the buttons actually works.
- This kind of prototype is sufficient to investigate scenarios of use and to decide, for example, whether the button images and labels are appropriate and the functions sufficient, but not to test whether the speech is loud enough or the response fast enough.
- Prototyping is either high-fidelity or low-fidelity in nature.

- Prototypes don’t necessarily look like final products they can have different fidelity.
- The fidelity of a prototype refers to how it conveys the look-and-feel of the final product (basically, its level of detail and realism).
- Fidelity can vary in the areas of: Visual design, Content, Interactivity

Low-Fidelity Prototyping

- A low-fidelity prototype does not look very much like the final product, nor does it provide the same functionality.
- For example, it may use very different materials, such as paper and cardboard rather than electronic screens and metal; it may perform only a limited set of functions; or it may only represent the functions and not perform any of them.

- The block of wood used to prototype the PalmPilot described earlier is a low-fidelity prototype.
- Low-fidelity prototypes are useful because they tend to be simple, cheap, and quick to produce.
- Low-fidelity prototyping is of lesser quality than the desired product.
- Low-fidelity prototyping entails a design sketch on physical materials like paper or board.

❖For example, you can perform low-fidelity prototyping by sketching the web pages and linking the button tabs to the pages when building a web page.

❖Low-fidelity prototyping is of low quality and also low cost.

❖It saves resources, but may not have every function that a website should have.

Common low fidelity prototypes

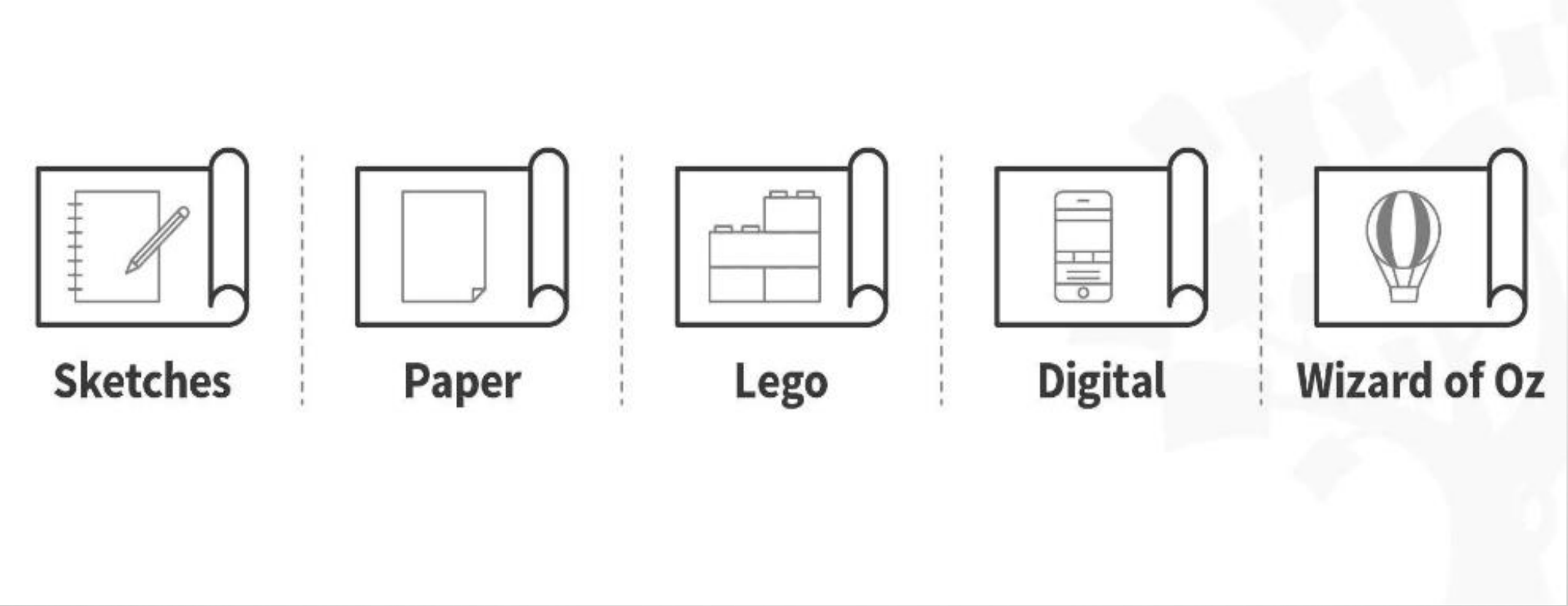

Figure 3: Types of low fidelity prototypes (The Interaction Design Foundation, 2023)

Sketches

- While sketches are often considered to not be technically prototypes, they can be extremely helpful for making decisions, mostly because they are incredibly easy to create and even easier to discard.
- We don’t need any artistic skill to sketch well, so this is a great tool for designers and non-designers alike.

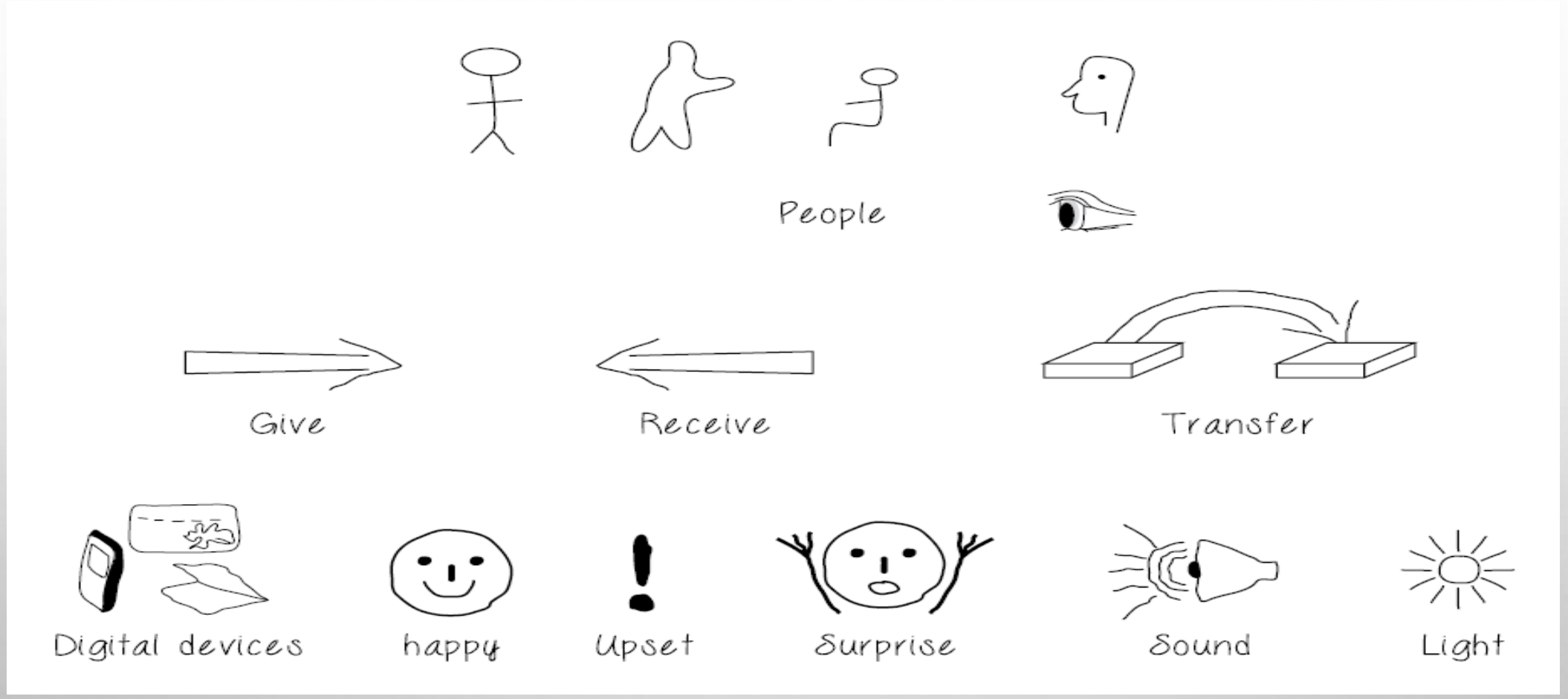

Figure 4: Some simple sketches for low-fidelity prototyping (Yvonne Rogers, 2023, P.448)

Pros and Cons of Sketches

Pros of sketches

- They are extremely cheap and fast to create. As such, you can sketch out a large number of ideas in a short amount of time.
- You can do it anywhere: with pen and paper or digitally on your smartphone, tablet or desktop computer.
- They are disposable, so you won't get attached to sketches that turn out to be bad ideas.

Cons of sketches

- Sketches lack detail and are ambiguous by design. As such, you cannot use sketches to convey complex interactions of an app.
- Sketches are almost never of high enough fidelity to be useful with people outside of the team, since they rarely have the context to understand what the sketch is meant to convey.
- Sketches are not very helpful in convergent processes where you want to select a few best ideas.

When to Use Sketches

- Use sketches in early, divergent stages of your design process.
- Sketch out your rough ideas so you can discuss them with team-mates.
- You can also sketch diagrams and mind maps in order to illustrate a system, process or the structure of your ideas.
- Sketch the touch points that affect a user's journey, and then identify how they relate to one another.

Paper prototypes

- In Paper prototypes, design teams create paper representations of digital products to help them realize concepts and test designs.
- They draw sketches or adapt printed materials and use these low-fidelity screenshot samples to cheaply guide their designs and study users' reactions from early in projects.

Pros and Cons of Paper Prototypes

Pros of Paper Prototypes

- Paper prototypes are cheap and easy to create as well as modify.
- You can ignore the deeper, superficial details of an interface, such as the color of a button can allows you to test the concept of your idea, rather than its visual execution.
- Paper prototypes are very obviously unfinished; therefore, users are unlikely to hold back their critiques for fear of hurting your feelings.

Cons of Paper Prototypes

- While generally easy to create, sometimes you might spend a bit of time to make a paper prototype. You might get emotionally attached as a result and become unable to objectively evaluate its merits.
- Paper prototypes are less helpful to test commonly used user interface patterns. That's because users are likely to already know how the user interface works.

- You can only test paper prototypes in person. Since the prototype is physical, you'll find it very difficult to conduct remote tests with it.
- While better than sketches, paper prototypes still require imagination from users. This means some users might struggle when they try to understand how the interface works.

When to Use Paper Prototypes

- Use paper prototypes when you're exploring novel solutions, to test whether people understand your solution.
- Don't use paper prototypes when you're revisiting the same solution, or using a standard user interface pattern to solve a problem.
- Use paper prototypes when you're exploring different ways of solving a problem. For instance, if you have different interface ideas to achieve the same user goal, you might want to sketch out a couple of different paper prototypes to test them on users.

Lego Prototypes

- Lego is a staple of any kid's toy box. Its versatility and ability to spark the imagination is what drives the company's success.
- As a designer, you can take advantage of Lego's versatility to create quick and simple prototypes of your ideas.

Pros and Cons of Lego Prototypes

Pros of Lego prototypes

- Lego allow you to quickly create physical prototypes you can build a rough model faster than most 3D printers can.
- Lego prototypes are versatile and easy to modify and dismantle. You can easily remove, add or rotate bricks to change your prototype.
- Lego prototypes encourage experimentation and fun, which are important components of success in the design thinking process.

Cons of Lego prototypes

- Lego prototypes are not suitable for digital products, such as mobile apps or websites.
- Lego prototypes are relatively expensive low-fidelity prototypes especially if you don’t have a set of Lego bricks.

When to Use Lego Prototypes

- Use Lego prototypes to empathize with your users. Use Lego bricks to recreate and reenact user journeys cheaply and visually.
- Use Lego prototypes when your solution involves a complex system of different parties.
- When you’re creating complex physical products, you can use Lego pieces to create quick and dirty prototypes.

Digital Wireframes

- Wireframes are simple, bare-bones illustrations of your app or website.
- They allow you to ignore the visual and interactive aspects of your prototype and focus on content structure and functionality.

Pros and Cons of Wireframes

Pros of wireframes

- You can quickly change your wireframes, compared with higher-fidelity prototypes such as app mockups. This is because wireframes don’t contain details such as images and colors.
- Wireframes let you focus on the functionality and content structure of the product. This means you can ignore visuals, such as colors and fonts, in favor of polishing the core functions of the app.
- Wireframes, let you communicate the relation between different pages in your product.

Cons of wireframes

- Since wireframes are still quite bare-bones, users might struggle to understand how what you present to them works.
- Wireframes have encouraged placeholder content, in the past. This is no longer advised, since copy and images that are significantly different from your placeholders will absolutely affect the final user experience.

When to Use Wireframes

- Use wireframes slightly later in your design process, when you are ready to flesh out a few design ideas.
- You should not use (digital) wireframes when you are in the divergent stages of your design process if they slow you down.
- Use wireframes when you are ready to think about topics such as how to create optimal user flows, what kinds of templates you should use for various screens and pages and how much space to allocate various elements on a screen.

Wizard of Oz Prototypes

- Wizard of Oz prototypes are prototypes with fake functions for instance, where you get a team-mate to mimic complex interactions rather than code a piece of software for it.
- The idea of Wizard of Oz prototypes is to get users to believe that the prototype is fully functional, so you can test it while saving time and resources.
- For example, you can create a Wizard of Oz prototype for a smart assistant, where your team-mate types out responses to trick the user into thinking that the smart assistant is fully functional.

Pros and Cons of Wizard of Oz Prototypes

Pros of Wizard of Oz prototypes

- You can test particularly complex parts of your design without having to build it. This allows you to validate your design before you spend more resources to implement it.
- You can test future technologies easily without building a complicated prototype. This allows you to fine-tune the requirements of the technology.
- Users tend to provide realistic feedback, since Wizard of Oz prototypes are more believable and interactive.

Cons of Wizard of Oz prototypes

- You'll need to spend some time to build your Wizard of Oz prototype. Since you need the user to believe that it's fully functional, you'll need to make it look convincingly polished.
- You have to train a "wizard" who'll simulate the responses of the system.
- The wizard might not act consistently throughout tests. Thus, your system might behave differently from test to test, which affects your test results.

When to Use Wizard of Oz Prototypes

- Use Wizard of Oz prototypes in the late stages of the design process.
- Use Wizard of Oz prototypes when you're designing complex systems or designing for future technologies.
- Wizard of Oz prototypes can also be extremely useful when prototyping any sort of voice interface or chat system where the backend would be hard to build but easy for a human to fake.

High-Fidelity Prototyping

- A high-fidelity prototype looks more like the final product and usually provides more functionality than a low-fidelity prototype.
- For example, a prototype of a software system developed in Python or other executable language is higher fidelity than a paper-based mock-up; a molded piece of plastic with a dummy keyboard would be a higher-fidelity prototype of the PalmPilot than the block of wood.

- Common strategy for developing a high-fidelity software prototype is to focus on the functions and not include any error handling, for example.
- There is a continuum between low- and high-fidelity, and prototypes used in the wild, for example, will have enough fidelity to be able to answer their design questions and to learn about interaction or technological constraints or contextual factors.

- One of the consequences of high-fidelity prototypes is that the prototype can appear to be good enough to be the final product, and stakeholders may be less prepared to critique it, or may critique it only superficially.
- To avoid this, it is important to focus on questions that prompt feedback on specific aspects whenever showing prototypes to stakeholders.
- Another consequence may be that fewer alternatives are considered because the prototype works and users like it.

- High-fidelity prototypes can be developed by modifying and integrating existing components both hardware and software which are widely available through various developer kits and open source software.

Compromises in Prototyping

- By their very nature, prototypes involve compromises: the intention is to produce something quickly to test an aspect of the product.
- The kind of questions that any one prototype can answer is limited, and the prototype must be built with the key issues in mind.
- In low-fidelity prototyping, it is fairly clear that compromises have been made.

- For example, with a paper-based prototype, an obvious compromise is that the device doesn’t actually work. For physical prototypes or software prototypes, some of the compromises will still be fairly clear.
- For example, the casing may not be very robust, the response speed may be slow, the look and feel may not be finalized, or only a limited amount of functionality may be available.
- Two common properties that are often traded off against each other are breadth of functionality versus depth.

- These two kinds of prototyping are called horizontal prototyping (providing a wide range of functions but with little detail) and vertical prototyping (providing a lot of detail for only a few functions).
- Another common compromise is level of robustness versus degree of changeability. Making a prototype robust may lead to it being harder to change.
- This compromise may not be visible until something goes wrong. For example, the internal structure of a piece of software may not have been carefully designed, or the connections between electronic components may be delicate.

- Although prototypes may have undergone extensive evaluation, they may not have been built with good engineering principles, or been subjected to rigorous quality testing for other characteristics such as security and error-free operation.
- Building a product to be used by thousands or millions of people running on various platforms and under a wide range of circumstances requires a different construction and testing regime than producing a quick prototype to answer specific questions.

Conceptual design

- Conceptual design is concerned with developing a conceptual model.
- The idea of a conceptual model can be difficult to grasp because these models take many different forms and there isn't a definitive detailed characterization of one.
- A conceptual model is an outline of what people can do with a product and which concepts are needed for them to understand how to interact with it.

- This emerges from an understanding of the problem space and the current functional requirements including information about people who might use the product, their context, and their goals.
- Interaction design often suggests that designers empathize with the people they are designing for, but sometimes trying to empathize with others is not the right approach.

- Different creativity and brainstorming techniques can be used to explore ideas within the design team, together with scenarios and personas.
- The availability of ready-made components increases the ease with which ideas can be prototyped, which also helps to explore different conceptual models and design ideas.

Developing an Initial Conceptual Model

- The core components of the conceptual model are metaphors and analogies, the concepts to which users are exposed, the relationship between those concepts, and the mappings between the concepts and user experience being supported.
- Some of these will derive from the product's requirements, such as the concepts involved in an activity and their relationships, which may be captured through scenarios and use cases.
- Others such as suitable metaphors and analogies will be informed by immersion in the data and understanding the application domain.

The following are the different approaches to produce an initial conceptual model, they provide different ways of thinking about the product and help generate potential conceptual models. :

- How to choose interface metaphors that will help users understand the product?
- Which interaction type(s) would best support the users' activities?
- Do different interface types suggest alternative design insights or options?

Interface Metaphors

- Interface metaphors combine familiar knowledge with new knowledge in a way that will help people understand the product.
- Choosing suitable metaphors and combining new and familiar concepts requires a balance between utility and relevance, and it is based on understanding users and their context.
- For example, consider an educational system to teach 6-year-olds mathematics. One possible metaphor is a classroom with a teacher standing at the front, but this may not appeal to all children in that age range.

- A metaphor that builds on something enjoyable for all 6-year-olds is more likely to keep them engaged, such as a ball game, the circus, a playroom, and so on.
- Different approaches to identifying and choosing an interface metaphor have been tried, and different factors may be considered.
- For example, combining creativity methods to explore everyday objects, paper prototypes, and toolkits to support groups of students designing novel interface metaphors and gestures for mobile devices.

Steps for choosing a good interface metaphors

- The first step is to understand what the system will do, by identifying functional requirements, developing partial conceptual models and trying them may be part of the process.
- The second step is to understand which bits of the product are likely to cause users problems i.e., which tasks or subtasks cause problems, are complicated, or are critical.
- Understanding areas in which users are likely to have difficulties means that the metaphor can be chosen to support those aspects.
- The third step is to generate metaphors.

- When suitable metaphors have been generated, they need to be evaluated through the following guiding questions:
- How much structure does the metaphor provide? A good metaphor will provide structure preferably familiar structure.
- How much of the metaphor is relevant to the problem? One of the difficulties of using metaphors is that users may think they understand more than they do and start applying inappropriate elements of the metaphor to the product, leading to confusion or false expectations.

- Is the interface metaphor easy to represent? A good metaphor will be associated with particular physical, visual, and audio elements, as well as words.
- Will your audience understand the metaphor?
- How extensible is the metaphor? Does it have extra aspects that may be useful later?

Interaction Types

- Earlier on, different types of interaction: instructing, conversing, manipulating, exploring, and responding.
- Which type of interaction is best suited to the current design depends on the application domain and the kind of product being developed.
- For example, a computer game is most likely to suit a manipulating style, while a software application for drawing or drafting has aspects of instructing and conversing.

- Most conceptual models will include a combination of interaction types, and different parts of the interaction will be associated with different types.
- For example, in the group travel organizer, one of the tasks is to find out the visa regulations for a particular destination.
- This will require an instructing approach to interaction as no dialog is necessary for the system to show the regulations.

Interface Types

- Considering different interfaces at this stage may seem premature, but it has both a design and a practical purpose.
- When thinking about the conceptual model for a product, it is important not to be unduly influenced by a predetermined interface type.
- Different interface types prompt and support different perspectives on potential user experiences and possible behaviors, hence prompting alternative design ideas.

- Which ones to choose depends on the product constraints that arise from the requirements. For example, input and output modes will be influenced by user and environmental requirements.
- Therefore, considering interfaces at this point also takes one step toward producing practical prototypes.

Expanding the Initial Conceptual Model

- Prototyping or conducting evaluations ideas need some expansion.
- Examples include which functions the product will perform and which the user will perform, how those functions are related, and what information is required to support them.
- Some of this will have been considered during the requirements activity and will evolve after prototyping and evaluation.

What Functions Will the Product Perform?

- This question is about whether the product or the user takes responsibility for different parts of the overall goal.
- For example, the travel organizer is intended to suggest specific vacation options for a group of people, but is that all it should do? It could automatically reserve the bookings. Or should it wait until it is given a preferred choice?

How Are the Functions Related to Each Other?

- Functions may be related temporally; for example, one must be performed before another, or two can be performed in parallel.
- They may also be related through any number of possible categorizations, for instance, all functions relating to privacy on a smartphone or all options for viewing photographs on a social networking site.

What Information Is Needed?

- What data is required to perform a task? How is this data to be transformed by the system? Data is one of the categories of requirements identified and captured through the requirements activity.
- Detailed issues of structure and display, such as whether to use an analog display or a digital display, will more likely be dealt with during the concrete design activity, but implications arising from the type of data to be displayed may impact conceptual design issues.

Concrete Design

- Conceptual design and concrete design are closely related.
- The difference between them is more a matter of changing emphasis: conceptual issues will sometimes be highlighted, while at other times concrete detail will be the focus.
- Producing a prototype inevitably means making some concrete decisions, though tentatively, and since interaction design is iterative, some detailed issues will come up during conceptual design, and vice versa.

- Design involves balancing the range of environmental, user, data, usability, and user experience requirements with functional requirements.
- For example, the functionality of a wearable interactive product may be constrained by the activities someone wants to perform while wearing it; a computer game may need to be learnable but also challenging.
- There are many aspects to the concrete design of interactive products: visual appearance such as colors and graphics, icon design, button or gestural design, navigation, layout, choice of interaction devices, and so on.

- Several interface types, together with their associated design considerations, guidelines, principles, and rules, which help designers ensure that their products meet usability and user experience goals.
- These represent the kinds of decision that are made during concrete design.
- Two aspects that have drawn particular attention for concrete design are accessibility and inclusiveness in order to empower people in their everyday and working lives.

- Accessibility considerations in concrete design include input and output modes.
- Apart from standard keyboard, mouse, and touchscreen, there are also different pointing devices and keyboards, screen readers, voice, sensors, and cameras, among others.
- Interactive products must be flexible enough to work with these various devices.

Generating prototypes

- Generating prototypes in interface design is the process of creating a working model of the user interface of a product or system to test its functionality and usability.
- Prototypes can be used to test different design ideas, get feedback from users and stakeholders, and make necessary changes before development begins.

ways to generate prototypes

- Sketching: Sketching is a quick and easy way to generate low-fidelity prototypes. Sketches can be used to test the overall layout of the user interface and the placement of individual elements.
- Wireframing: Wireframes are more detailed prototypes that show the placement of all elements on the user interface, as well as their size and relationship to each other.

- Mockups: Mockups are high-fidelity prototypes that look and feel like the final product. Mockups can be created using design software, such as Photoshop or Sketch.
- Interactive prototypes: Interactive prototypes allow users to interact with the prototype as if it were the final product.

How to generate effective prototypes

- Start with a clear understanding of the purpose of the prototype. What do you want to learn from the prototype? Once you know the purpose, you can choose the right method and level of fidelity.
- Get feedback early and often. Share your prototypes with users and stakeholders early on to get feedback and identify potential problems.
- Use iterative prototyping. Create a simple prototype, test it with users, and then use the feedback to refine the prototype. This process can be repeated until you are satisfied with the results.

Examples of how to generate prototypes

- Generating a prototype for a new mobile app: You could start by sketching the layout of the main screens of the app.
- Once you are happy with the layout, you could create wireframes to show the placement of all elements on the screens.
- Finally, you could create a mockup or interactive prototype to test the usability of the app and get feedback from users.

- Generating a prototype for a new website: You could start by creating a sitemap to show the structure of the website and the hierarchy of the pages.
- Once you have a sitemap, you could create wireframes for each page of the website.
- Finally, you could create a mockup or interactive prototype to test the usability of the website and get feedback from users.

Construction

- Construction in interface design refers to the process of creating and arranging the elements of a user interface (UI) in a way that is both visually appealing and easy to use.
- It involves creating a hierarchy of information, using visual cues to guide users, and ensuring that all elements are aligned and consistent.
- As prototyping and building alternatives progresses, development will focus on higher-fidelity prototypes and developing the final product.

- This is facilitated by putting together readymade components, such as a set of alarms, sensors, and lights to make a physical product, or code libraries to generate a piece of software, or both.
- Whatever the final form, it is unlikely that anything will need to be developed from scratch, as there are many useful (in some cases essential) resources to support development.

Key principles of construction

- **Hierarchy:** The most important information on a page should be the most visually prominent. This can be achieved by using larger fonts, brighter colors, and bolder text.
- **Grouping**: Related elements should be grouped together to make them easier to scan and understand. This can be done using boxes, borders, or white space.
- **Alignment:** All elements on a page should be aligned to create a sense of order and consistency. This can be done using horizontal grids, vertical grids, or baselines.

- **Proximity**: Closely related elements should be placed close together, while less related elements should be placed further apart. This helps users to understand the relationship between different elements on a page.
- **Balance:** The elements on a page should be balanced in terms of their visual weight. This means that no one area of the page should be too overwhelming or too empty.

Examples of how construction is used

- The main navigation menu of a website is typically placed at the top of the page. This is because it is one of the most important elements on the page, and users should be able to easily find it.
- Related product items on an e-commerce website are typically grouped together. This makes it easier for users to compare and contrast products, and to find the products they are looking for.

- Form elements on a website are typically aligned with each other. This makes it easier for users to fill out the form, and to avoid making mistakes.
- Call-to-action buttons on a website are typically placed prominently on the page. This is because they are one of the most important elements on the page, and users should be able to easily find them.

References

- Preece, R., Rogers, Y., & Sharp, H. (2023). Interaction Design: Beyond Human-Computer Interaction. J. Wiley & Sons.
- The Interaction Design Foundation(2023). 5 Common Low-Fidelity Prototypes and Their Best Practices. https://www.interaction-design.org/literature/article/prototyping-learn-eight-common-methods-and-best-practices#:~:text=Here%2C%20let's%20look%20at%20the,and%20Wizard%20of%20Oz%20prototypes.

Thank you

Next Lecture We Shall Look At

Prototyping and construction: generating prototypes, construction